

Nummernmerker

Tassilo Tanneberger

November 24, 2019

1 Ideen der Algorithmen

Ist eine Sammlung an prinzipiellen Gedanken, die ich mir gemacht habe bevor ich angefangen habe zu implementieren.

1.1 Dreier Blöcke

Ich habe mich dafür entschieden als erstes die Zahl in dreier Blöcke zu spalten. Das hatte den einfachen Grund, dass man somit später flexibler ist, wenn man Zahlen umherschreibt. Ohne Probleme kann man dann z.B. eine Zahl von dem vorherigen Block abtrennen und an seinen anfügen, ohne die Grenzen von $2 \leq x \leq 4$ zu überschreiten.

1.2 Anfügen

Wird dann durchgeführt, wenn die letzte Zahl des davor liegenden Blocks keine 0 ist. Dabei wird die letzte Zahl nun abgetrennt und an den aktuellen Block angefügt.

before = "123", "012"
after = "12", "3012"

1.3 Abfügen

Wird dann durchgeführt, wenn sowohl die letzte Zahl des davor liegenden Gliedes eine 0 ist. Dabei wird nun die erste Zahl aus dem aktuellen Glied abgetrennt und an das Letzte des Vorherigen angefügt.

before = "120", "012"
after = "1200", "12"

2 Implementierung

Das Programm wurde in Go implementiert. Go ist obwohl noch kompiliert werden muss und es relativ low level ist, noch eine sehr angenehme Sprache und ideal für solche kleine Projekte.

2.1 Transform Blocks

Diese Funktion ist der Kern des Programmes. Das Programm iteriert durch die Blöcke und schaut , ob das erste Zeichen eine "0" ist. Ist das der Fall, wird geschaut ob das letzte Zeichen des vorherigen Blockes als letzten Charakter eine "0" hat. Ist das der Fall, wird Abfügen durchgeführt wenn es nicht der Fall ist Anfügen.

2.2 Finishing up

2.3 Benchmarking

Ich hatte ein paar Versuche mit zufällig generierten Zahlen gemacht. Das sind die Resultate:

- "Anfügen" aktiv: 82 %
- "Anfügen" und "Abfügen" aktiv: 98 %
- aller Anfangs 0. Können beseitigt werden.

3 Building

Es ist wieder empfohlen Linux zu nutzen.

```
$ go build -o Nummernmerker main.go
```

```
$ ./Nummernmerker
```